

**L
N**

LN
LN
LNLN
LN
LN

LN
LN
LN
LN
LN
LN

LN

LN

LN
LNLN
LNLN
LNLN
LNLN
LNLN
LN

LN

LN

LN

LN
LNLN
LN

LN

100

100

```
LL      NN      NN  KK      KK  PPPPPPPP  RRRRRRRR  000000  SSSSSSSS  HH      HH  RRRRRRRR
LL      NN      NN  KK      KK  PPPPPPPP  RRRRRRRR  000000  SSSSSSSS  HH      HH  RRRRRRRR
LL      NN      NN  KK      KK  PP      PP  RR      RR  00      00  SS      SS  HH      HH  RR      RR
LL      NN      NN  KK      KK  PP      PP  RR      RR  00      00  SS      SS  HH      HH  RR      RR
LL      NNNN     NN  KK      KK      PP      PP  RR      RR  00      00  SS      SS  HH      HH  RR      RR
LL      NNNN     NN  KK      KK      PP      PP  RR      RR  00      00  SS      SS  HH      HH  RR      RR
LL      NN  NN  NN  KKKKKK  PPPPPPPP  RRRRRRRR  00      00  SSSSSS  HHHHHHHHHH  RRRRRRRR
LL      NN  NN  NN  KKKKKK  PPPPPPPP  RRRRRRRR  00      00  SSSSSS  HHHHHHHHHH  RRRRRRRR
LL      NN      NNNN  KK      KK  PP      PP  RR      RR  00      00      SS  HH      HH  RR      RR
LL      NN      NNNN  KK      KK  PP      PP  RR      RR  00      00      SS  HH      HH  RR      RR
LL      NN      NN  KK      KK      PP      PP  RR      RR  00      00      SS  HH      HH  RR      RR
LL      NN      NN  KK      KK      PP      PP  RR      RR  00      00      SS  HH      HH  RR      RR
LLLLLLLLLL  NN      NN  KK      KK  PP      PP  RR      RR  000000  SSSSSSSS  HH      HH  RR      RR
LLLLLLLLLL  NN      NN  KK      KK  PP      PP  RR      RR  000000  SSSSSSSS  HH      HH  RR      RR
                                           ....
                                           ....
                                           ....
                                           ....
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

```

0001 0 module lnk_procshrim ! PROCESS SHAREABLE IMAGES ON PASS 1
0002 0 (ident = 'V04-000'
0003 0 ,addressing_mode
0004 0 (external = general
0005 0 ,nonexternal = long_relative
0006 0 )
0007 0 ) =
0008 1 begin
0009 1
0010 1 *****
0011 1 *
0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0014 1 * ALL RIGHTS RESERVED.
0015 1 *
0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0021 1 * TRANSFERRED.
0022 1 *
0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0025 1 * CORPORATION.
0026 1 *
0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0029 1 *
0030 1 *****
0031 1
0032 1
0033 1
0034 1 ++
0035 1 FACILITY: LINKER
0036 1
0037 1 ABSTRACT: THIS MODULE CONTAINS THE ROUTINES TO READ SHAREABLE
0038 1 IMAGES ON PASS 1.
0039 1
0040 1
0041 1 ENVIRONMENT: VMS NATIVE MODE
0042 1
0043 1 AUTHOR: T.J. PORTER, CREATION DATE: 15-DEC-77
0044 1
0045 1 MODIFIED BY:
0046 1
0047 1 V03-008 ADE0003 Alan D. Eldridge 7-Aug-1984
0048 1 Propagate the DZRO bit from the shareable image ISD to the
0049 1 corresponding ISD in the new image. It was being cleared
0050 1 unconditionally.
0051 1
0052 1 V03-007 JWT0192 Jim Teague 2-Aug-1984
0053 1 Unfortunately, push came to shove, and the global
0054 1 isd search optimization has been tossed.
0055 1
0056 1 V03-006 JWT0189 Jim Teague 25-Jul-1984
0057 1 Make some changes to accomodate demand-zero image

```



```
58      0058 1 | sections in shareable images. The minimum image
59      0059 1 | file VBN for the symbol table is now 2, instead of 3.
60      0060 1 | Why? Consider the case of a shareable image with
61      0061 1 | nothing BUT a demand zero image section...
62      0062 1 |
63      0063 1 | V03-005 ADE0002      Alan D. Eldridge      10-Jul-1984
64      0064 1 | Fix null arguments passed in SIGNAL due to editing errors.
65      0065 1 |
66      0066 1 | V03-004 ADE0001      Alan D. Eldridge      26-Feb-1984
67      0067 1 | Don't erase CLUSL_GSMATCH. It is no longer re-used as
68      0068 1 | something else in pass 2, and it is referenced to correctly
69      0069 1 | perform the GSMATCH checking in pass one.
70      0070 1 |
71      0071 1 | V03-003 JWT0152      Jim Teague      8-Feb-1984
72      0072 1 | Long names for global image section descriptors.
73      0073 1 |
74      0074 1 | V03-002 JWT0111      Jim Teague      14-Apr-1983
75      0075 1 | Don't pull in shareable images that are in global
76      0076 1 | isds unless they're based.
77      0077 1 |
78      0078 1 | V03-001 JWT0044      Jim Teague      30-Jul-1982
79      0079 1 | Open file performance boost.
80      0080 1 |
81      0081 1 | --
82      0082 1 |
83      0083 1 |
84      0084 1 | TABLE OF CONTENTS:
85      0085 1 |
86      0086 1 | forward routine
87      0087 1 | readnextblock;
88      0088 1 |
89      0089 1 |
90      0090 1 | INCLUDE FILES:
91      0091 1 |
92      0092 1 | library 'LIBL32';
93      0093 1 | require 'PREFIX';
94      0208 1 | library 'DATBAS';
95      0209 1 | require 'ISGENC';
96      0593 1 |
97      0594 1 |
98      0595 1 | MACROS:
99      0596 1 |
100     0597 1 | NONE
101     0598 1 |
102     0599 1 | EQUATED SYMBOLS:
103     0600 1 |
104     0601 1 | NONE
105     0602 1 |
106     0603 1 | EXTERNAL REFERENCES:
107     0604 1 |
108     0605 1 |
109     0606 1 | external routine
110     0607 1 | lnk$alloblk      : novalue,
111     0608 1 | lnk$alloccluster,
112     0609 1 | lnk$closefile    : novalue,
113     0610 1 | lnk$pointobj     : novalue,
114     0611 1 | lnk$addimage,
```

! READS THE NEXT HEADER BLOCK OF IMAGE

! VMS SYSTEM STRUCTURE DEFINITIONS
! USEFUL GENERAL MACROS
! INTERNAL DATA BASE DEFINITIONS
! IMAGE SECTION PARAMETERS! DYNAMIC MEMORY ALLOCATOR
! ALLOCATE CLUSTER DESCRIPTOR
! CLOSE CURRENT INPUT FILE
! POINTS TO NEW PLACE IN FILE
! PROCESS SHAREABLE IMAGE


```

: 115      0612 1      lnk$procsobj;          ! PROCESSES OBJECT MODULES (I.E. THE GST)
: 116      0613 1
: 117      0614 1      external literal
: 118      0615 1      lnk$_badimgHdr,        ! BAD IMAGE HEADER ERROR MESSAGE
: 119      0616 1      lnk$_basshrbel,        ! BASED SHAREABLE IMAGE BELOW BASE=
: 120      0617 1      lnk$_confbasadr,      ! CONFLICTING BASE ADDRESSES FOR SHR IMGS
: 121      0618 1      lnk$_idmismch,        ! GSMATCH MISMATCH WITH SHR IMG STB LIBRARY
: 122      0619 1      lnk$_imgbased,        ! ATTEMPT TO RE-BASE A NON-PIC IMAGE
: 123      0620 1      lnk$_noimgfil,        ! NO IMAGE FILE CREATED
: 124      0621 1      lnk$_nonpicimg,       ! SHAREABLE IMAGE IS NON-PIC
: 125      0622 1      lnk$_relink,         ! RELINK DUE TO COPYALWAYS SECTION
: 126      0623 1      lnk$_readerr;        ! READ ERROR
: 127      0624 1
: 128      0625 1      external
: 129      0626 1      lnk$gl_shrcIstrs,      ! COUNT OF NUMBER OF SHAREABLE IMAGE CLUSTERS (PIC AND NON-P
: 130      0627 1      lnk$gl_rab           : block [, byte],      ! RAB FOR OPEN IMAGE FILE
: 131      0628 1      lnk$gl_nisects       : word,                ! IMAGE SECTION ACCUMULATOR
: 132      0629 1      lnk$gl_shriscts      : word,                ! NUMBER OF SHAREABLE IMAGE ISECTS
: 133      0630 1      lnk$gl_pass          : byte,                ! CURRENT PASS
: 134      0631 1      lnk$gl_curfil        : ref block [, byte],  ! CURRENT FILE POINTER
: 135      0632 1      lnk$gl_curclu        : ref block [, byte],  ! CURRENT CLUSTER DESCRIPTOR
: 136      0633 1      lnk$gl_defclu        : block [, byte],     ! DEFAULT CLUSTER DESCRIPTOR
: 137      0634 1      lnk$gl_lastclu       : ref block [, byte],  ! POINTER TO LAST CLUSTER DESCRIPTOR
: 138      0635 1      lnk$gl_ctlmsk       : block [, byte],     ! CONTROL FLAGS
: 139      0636 1      lnk$gl_objrecs ;      ! COUNT OF OBJECT RECORDS
: 140      0637 1
: 141      0638 1      !
: 142      0639 1      ! MODULE OWN STORAGE:
: 143      0640 1      !
: 144      0641 1      global
: 145      0642 1      lnk$gl_gsbufdsc      : vector [3],         ! BUFFER DESCRIPTOR FOR COPIED GLOBAL SECTIONS
: 146      0643 1      lnk$gl_imgrecs ;     ! COUNT OF IMAGE RECORDS
: 147      0644 1
: 148      0645 1      own
: 149      0646 1      curisdseq            : vector [4, byte]    ! SUFFIX TO SCLUSTER NAME GIVING GBL ISD NAME
: 150      0647 1      initial (%ascii '_000'),
: 151      0648 1      hdrblkcnt,          ! NUMBER REMAINING HEADER BLOCKS
: 152      0649 1      headerblock ;       ! CURRENT HEADER VBN
: 153      0650 1
: 154      0651 1
```



```
156 0652 1 global routine lnk$procshrim (modrfa) = !
157 0653 1 ++
158 0654 1 FUNCTIONAL DESCRIPTION:
159 0655 1 THIS ROUTINE IS CALLED TO PROCESS SHAREABLE IMAGES ON PASS
160 0656 1 ONE OF THE LINK. IT READS AND VALIDATES EVERY BLOCK OF THE IMAGE HEADER
161 0657 1 BUILDING THE LIST OF IMAGE SECTION DESCRIPTORS. AFTER COMPLETING THE HEADER
162 0658 1 IT SETS THE RAB POINTING TO THE SYMBOL TABLE PART OF THE IMAGE FILE AND
163 0659 1 CALLS LNK$PROCSOBJ TO DO THE PASS ONE OBJECT MODULE PROCESSING
164 0660 1 OF THE SYMBOL TABLE.
165 0661 1 THE SYMBOL TABLE OF A SHAREABLE IMAGE CONTAINS ALL THE UNIVERSAL
166 0662 1 SYMBOLS DEFINED FOR THE IMAGE AND IS IN THE OBJECT MODULE FORMAT.
167 0663 1
168 0664 1 FORMAL PARAMETERS:
169 0665 1 MODRFA IS A POINTER TO THE 6 BYTE RFA OF THE SHAREABLE IMAGE IF
170 0666 1 IT IS IN A LIBRARY. IF NOT A LIBRARY SHAREABLE IMAGE THE ARGUMENT
171 0667 1 DOES NOT EXIST.
172 0668 1
173 0669 1 IMPLICIT INPUTS:
174 0670 1 SEE ABOVE EXTERNAL DECLARATIONS.
175 0671 1 IN ADDITION THE FILE CONTAINING THE IMAGE IS ALREADY OPEN
176 0672 1 FOR BLOCK READ OPERATIONS.
177 0673 1
178 0674 1 IMPLICIT OUTPUTS:
179 0675 1 SEE ABOVE GLOBAL DECLARATIONS.
180 0676 1 IN ADDITION ALL IMAGE SECTIONS FOUND IN THIS SHAREABLE IMAGE
181 0677 1 ARE APPENDED TO THE LIST AND THE GST HAS BEEN PROCESSED AS AN OBJECT
182 0678 1 MODULE. THAT IS ALL UNIVERSAL SYMBOLS ARE IN THE LINKER SYMBOL TABLE.
183 0679 1
184 0680 1 ROUTINE VALUE:
185 0681 1 RETURNS VALUE TRUE IF SUCCESSFULLY PROCESSED, ELSE FALSE
186 0682 1
187 0683 1 SIDE EFFECTS:
188 0684 1 THE ROUTINE DOES NOT RETURN IF A FATAL ERROR IS DETECTED.
189 0685 1
190 0686 1 --
191 0687 2 begin
192 0688 2 builtin
193 0689 2 local actualcount ; ! GETS COUNT OF ARGUMENTS
194 0690 2
195 0691 2 nextisdoft, ! OFFSET TO NEXT ISD
196 0692 2 blockoffset, ! OFFSET IN FILE TO FIRST HEADER BLOCK
197 0693 2 isectident, ! MAJOR AND MINOR ID FROM HEADER
198 0694 2 firstisdvpg, ! VPG OF FIRST ISECT
199 0695 2 symdbgdatdsc : ref block [, byte], ! POINTER TO SYMBOL TABLE DESCRIPTOR
200 0696 2 gstrecs, ! NUMBER OF RECORDS IN SYMBOL TABLE
201 0697 2 ownclu : ref block [, byte], ! POINTER TO OWNING CLUSTER DESCRIPTOR
202 0698 2 ownfdb : ref block [, byte], ! AND IT'S FILE DESCRIPTOR BLOCK
203 0699 2 cludesc : ref block [, byte], ! POINTER TO CREATED CLUSTER DESCRIPTOR
204 0700 2 curhdrisd : ref block [, byte], ! POINTER TO CURRENT ISD IN HEADER
205 0701 2 curisd : ref block [, byte], ! POINTER TO CURRENT ISD BEING BUILT
206 0702 2 newhdrisd : ref block [, byte], ! POINTER TO HEADER PART OF CURRENT ISD BEING BUILT
207 0703 2 firstisd : ref block [, byte], ! POINTER TO FIRST ISECT IN CLUSTER
208 0704 2 gstvbn : vector [2, [long], ! RFA OF GST
209 0705 2 tafva, ! RELATIVE VA OF FIXUP SECTION
210 0706 2 saverecount, ! SAVED RECORD COUNT WHILE IN OBJPS1
211 0707 2 maxisdvbn ; ! LAST IMAGE VBN + 1
212 0708 2
```



```
213 0709 2 map
214 0710 2     modrfa : ref block [, byte] ;
215 0711 2
216 0712 2 bind
217 0713 2     header = lnk$al_rab [rab$l_ubf] : ref block [, byte] ;
218 0714 2
219 0715 2 if actualcount () eql 0
220 0716 2 then
221 0717 2     blockoffset = 0
222 0718 2
223 0719 2 else
224 0720 2     blockoffset = .modrfa [rfa$l_vbn] - 1 ;
225 0721 2
226 0722 2 lnk$gl_shrclstrs = .lnk$gl_shrclstrs + 1 ;
227 0723 2
228 0724 2 cludesc      = 0 ;
229 0725 2 hdrblkcnt    = 1 ;
230 0726 2 nextisdoff   = -1 ;
231 0727 2 maxisdvbn    = 0 ;
232 0728 2 headerblock = .blockoffset ;
233 0729 2
234 0730 2 ch$fill ('0', 3, curisdseq [1]) ;
235 0731 2
236 0732 2 while .nextisdoff eql -1
237 0733 2 do begin
238 0734 2     if not readnextblock ()
239 0735 2     then
240 0736 2         signal_stop (lin$badimgHdr, 1
241 0737 2             , lnk$gl_curfil [fdb$q_filename]
242 0738 2             )
243 0739 2     if .headerblock eql (.blockoffset + 1)
244 0740 2     then
245 0741 2         begin
246 0742 2             VALIDATE IMAGE HEADER
247 0743 2
248 0744 2             if .header [ihd$b_imgtype] neq ihd$k_shr
249 0745 2             or .header [ihd$w_majorid] neq ihd$k_majorid
250 0746 2             or ((.header [ihd$w_minorid]) < 0, 8, 0) eql (ihd$k_minorid and %x'FF')
251 0747 2             and
252 0748 2             (.header [ihd$w_minorid]) < 8, 8, 0) gtru (ihd$k_minorid and %x'FF00')/256
253 0749 2             )
254 0750 2             or .header [ihd$w_size] gtru maxu ((.header [ihd$w_patchoff]
255 0751 2                 + ihp$k_length)
256 0752 2                 , ihd$k_maxlength
257 0753 2             )
258 0754 2             or (hdrblkcnt = .header [ihd$b_hdrblkcnt] - 1) lss 0
259 0755 2             or (symdbgdtdsc = .header + .header [ihd$w_symdbgoff])
260 0756 2             gequ (.header + .header [ihd$w_size])
261 0757 2             or (gstrecs = .symdbgdtdsc [ihd$w_gstrecs]) lssu 3
262 0758 2             or (gstvbn [0] = .symdbgdtdsc [ihd$w_gstvbn])
263 0759 2             lequ (.hdrblkcnt + 1)
264 0760 2         then
265 0761 2             signal_stop (lin$badimgHdr, 1
266 0762 2                 , lnk$gl_curfil [fdb$q_filename]
267 0763 2                 )
268 0764 2             if not (lnk$gl_curclu [clu$v_pic] = .header [ihd$v_picimg])
269 0765 2             then
```

! POINTER TO BLOCK

! POINTER TO BLOCK BUFFER

! IF CALLED WITH NO ARGUMENTS

! HEADER AT START OF FILE

! OTHERWISE GET OFFSET,

! COUNT THIS SHAREABLE IMAGE CLUSTER

! NO CREATED CLUSTER YET
! MUST BE AT LEAST ONE BLOCK
! NEXT ISD IS ON NEXT BLOCK
! RESET LAST BLOCK OF IMAGE
! AND SET FOR FIRST BLOCK READ

! INITIALIZE THE ISD NAME SUFFIX

! WHILE THERE ARE MORE
! HEADER BLOCKS
! GET THE NEXT ONE

! AND IF UNSUCCESSFUL, FATAL IMAGE

! IF THE FIRST

! HAVE SOME VALIDATION TO DO

! CHECK IT IS A SHAREABLE IMAGE
! MAJOR HEADER ID MUST MATCH! THE HEADER FIXED PART
! MUST BE LESS THAN A BLOCK AND MUST! CONTAIN PATCH AREA. 0 TO
! 127 MORE BLOCKS. GET THE
! GST DESCRIPTOR WHICH MUST BE CONTAINED
! IN HEADER. MUST BE AT LEAST 3 RECORDS
! AND MUST BE BEYOND THE HEADER BLOCKS! ANY ABOVE NOT TRUE, FATAL IMAGE
! HEADER ERROR

! EXTRACT THE PIC BIT AND IF NON-PIC

```
270 0766 5 begin
271 0767 5 lnk$gl_ctlmsk [lnk$v_picing] = false ;
272 0768 5
273 0769 5 if .lnk$gl_curclu [clu$v_usrbased]
274 0770 5 and not .lnk$gl_curclu [clu$v_intclu]
275 0771 5 then
276 0772 6 begin
277 0773 6 signal (lin$imbased, 1
278 0774 6 ,lnk$gl_curfil [fdb$q_filename]
279 0775 6 ) ;
280 0776 6 lnk$gl_curclu [clu$v_usrbased] = false ;
281 0777 6 lnk$gl_curclu [clu$l_usrbase] = 0 ;
282 0778 5 end ;
283 0779 5 end
284 0780 4 else
285 0781 4 if .lnk$gl_curclu [clu$v_usrbased]
286 0782 4 then
287 0783 4 lnk$gl_ctlmsk [lnk$v_picing] = false ;
288 0784 4
289 0785 4 if
290 0786 4 and not .lnk$gl_ctlmsk [lnk$v_shr]
291 0787 4 and not .lnk$gl_ctlmsk [lnk$v_ubased]
292 0788 4 and not .lnk$gl_curclu [clu$v_pic]
293 0789 4 then
294 0790 4 signal (lin$nonpicing, 1
295 0791 4 ,lnk$gl_curfil [fdb$q_filename]
296 0792 5 ) ;
297 0793 4 if (lnk$gl_curclu [clu$v_matchctl] =
298 0794 4 .header [ihd$v_matchctl]) eql isd$k_matnev
299 0795 4 then
300 0796 4 lnk$gl_curclu [clu$v_copy] = true ;
301 0797 4 isectident = .header [ihd$l_ident] ;
302 0798 4 iafva = 0 ;
303 0799 4
304 0800 5 if (.header + .header [ihd$w_activoff])
305 0801 4 gtru header [ihd$l_iafva]
306 0802 4 then
307 0803 5 begin
308 0804 5 iafva = .header [ihd$l_iafva] ;
309 0805 5 if .iafva eql 0
310 0806 5 then
311 0807 5 lnk$gl_curclu [clu$v_prefixup] = true ;
312 0808 5 end
313 0809 4 else
314 0810 4 lnk$gl_curclu [clu$v_prefixup] = true ;
315 0811 4
316 0812 4 lnk$gl_imgrecs = .lnk$gl_imgrecs + .hdrblkcnt + 1 ;
317 0813 4 curhdrisd = .header + .header [ihd$w_size] ;
318 0814 4
319 0815 4
320 0816 4 CHECK GSMATCH OF IMAGE AGAINST GSMATCH FOUND IN SHAREABLE IMAGE
321 0817 4 LIBRARY IF THIS IMAGE FOUND IN A LIBRARY
322 0818 4
323 0819 4 if .lnk$gl_curclu [clu$l_gsmatch] neq 0
324 0820 4 and .lnk$gl_curclu [clu$l_gsmatch] neq .isectident
325 0821 4 then
326 0822 5 begin
```

```
! THIS IMAGE IS ALSO NON-PIC
! IF NON-PIC IMAGE BASED BY USER
! AND NOT AN INTERNALLY CREATED CLUSTER
! THEN THAT'S AN ERROR, BECAUSE WE CAN'T
! RELOCATE IT
! BUT IF CLUSTER BASED BY OPTION
! IF CREATING A SHAREABLE IMAGE
! AND IMAGE NOT BASED BY USER
! AND THIS IMAGE JUST MADE IT NON-PIC
! THEN TELL USER ABOUT IT (INFO)
! EXTRACT THE GLOBAL SECTION MATCH CONTROL
! MAKING PRIVATE COPY IF MATCH NEVER
! NOW THE IDENT TO WHICH IT APPLIES
! IF IMAGE HAS FIXUP VA OFFSET
! THEN PICK IT UP
! FIXVA THERE, BUT 0, SO ASSUME NOT THERE
! NO FIXUP VA, FLAG OLD IMAGE
! COUNT HEADER BLOCKS AS RECORDS
! POINT TO FIRST ISD IN HEADER
! IF FOUND IN SHAREABLE IMAGE LIB
! AND IMAGE/LIBRARY MISMATCH
```



```
bind libident = lnk$gl_curclu [clu$l_gsmatch] : block [, byte],
imgident = isectident : block [, byte];

signal (lin$_idmismch, 6 ! WARN USER ABOUT IDENT PROBLEM
, imgident [gmt$b_majorid], imgident [gmt$b_minorid]
, lnk$gl_curfil [fdb$q_filename], libident [gmt$b_majorid], libident [gmt$b_minorid]
, lnk$gl_curfil [fdb$q_libnamdsc]
);

end ;
else
curhdrisd = .header ;
! ALL DONE WITH FIRST BLOCK
! OTHER BLOCKS ARE ALL ISD

!
! PROCESS IMAGE SECTION DESCRIPTORS
while (nxtisdooff = (.curhdrisd [isd$w_size]) < 0, 16, 1) > 0 ! WHILE MORE ISD'S ON THIS BLOCK
do
begin
! BEGIN LOOP THAT PROCESSES EACH
if .curhdrisd [isd$b_type] neq isd$k_usrstack
then
begin
! IGNORING STACK SECTION
! IF THIS IS COPY ALWAYS SECTION
then
signal_stop (lin$_relink, 2 ! THEN ISSUE MESSAGE AND GIVE UP NOW
, lnk$gl_curfil [fdb$q_filename]
, lnk$gl_curfil [fdb$q_filename]
);

if .curhdrisd [isd$v_gbl] ! IF GLOBAL ISECT
then
begin
! THEN CHECK FROM SAME SHAREABLE IMAGE
local found,
shrdesc : block [dsc$c_s_bln, byte] ;
bind
gblnam = curhdrisd [isd$t_gblnam] : vector [, byte] ;

shrdesc [dsc$a_pointer] = gblnam [1] ; ! COMPLETE SECTION NAME DESCRIPTOR
shrdesc [dsc$w_length] = .gblnam [0] - suffix_size ; ! FIND LENGTH OF GLOBAL SECTION
! SECTION NAME IN ISECT DESCRIPTOR
if .cludesc eql 0 ! IF NO CREATED CLUSTERS YET
or not ch$eql (.shrdesc [dsc$w_length], gblnam [1] ! OR THIS IS FOR A DIFFERENT CLUSTER
, shrdesc [dsc$w_length], cludesc [clu$t_name]
)
then
begin
lnk$addimage (shrdesc, 0, cludesc, found) ; ! ADD IMAGE TO THE CLUSTER LIST
if .cludesc neq 0 and not .found
then
begin
cludesc [clu$v_prefixup] = .lnk$gl_curclu [clu$v_prefixup] ;
cludesc [clu$l_ownclu] = .lnk$gl_curclu ; ! COPY PREFIXUP FLAG FROM PARENT CLUSTER
! SET OWNING CLUSTER DESCRIPTOR
if not .lnk$gl_curclu [clu$v_pic] ! IF THIS IMAGE IS NON-PIC
then
begin
cludesc [clu$v_based] = true ; ! FLAG CLUSTER AS BASED
```

```
cludesc [clu$l_base] = (if .curhdrisd [isd$v_based]
                        then .curhdrisd [isd$v_vpg]^9
                        else .lnk$gl_curclu [clu$l_base] +
                            .lnk$gl_curclu [clu$l_cluoff]
                        ) ;
lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
                              .curhdrisd [isd$w_pagcnt]*512 ;
end ;
else
begin
if (.lnk$gl_curclu [clu$v_based]
    and not
    .lnk$gl_curclu [clu$v_pic]
)
or .lnk$gl_curclu [clu$v_usrbased]
then
begin
ownclu = .cludesc [clu$l_owncclu] ;          ! GET OWNING CLUSTER POINTER
if .ownclu neq 0
then
    lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
                                    .curhdrisd [isd$w_pagcnt]*512 ;

if .ownclu neq 0
    and
    .ownclu [clu$v_based]
    and
    .cludesc [clu$l_base] neq .curhdrisd [isd$v_vpg]^9
    ! IF THERE IS AN OWNING CLUSTER
    ! WHICH IS ALSO BASED
    ! AND WANTS THIS CLUSTER AT A
    ! DIFFERENT PLACE
then
begin
ownfdb = .ownclu [clu$l_fstfdb] ;          ! GET FIRST FILE DESCRIPTOR BLOCK
signal (lin$confbasadr, 5
        , cludesc [clu$b_namlng]
        , .curhdrisd [isd$v_vpg]^9, lnk$gl_curfil [fdb$g_filename]
        , .cludesc [clu$l_base], ownfdb [fdb$g_filename]
        , lin$_noimgfil
        ) ;
lnk$gl_ctlmsk [lnk$v_image] = false ;      ! DON'T MAKE A NON-RUNNABLE IMAGE
end
else
begin
if not .lnk$gl_curclu [clu$v_usrbased]
then
begin
if not .cludesc [clu$v_based]
    then
        lnk$gw_shriscts = .lnk$gw_shriscts - .cludesc [clu$l_nisects] ;

cludesc [clu$v_based] = true ;              ! FLAG AS BASED
cludesc [clu$l_base] = .curhdrisd [isd$v_vpg]^9 ;
end ;
end ;
end ;
```

```
384 0880 10
385 0881 10
386 0882 10
387 0883 10
388 0884 9
389 0885 9
390 0886 9
391 0887 8
392 0888 8
393 0889 7
394 0890 8
395 0891 9
396 0892 9
397 0893 9
398 0894 9
399 0895 8
400 0896 8
401 0897 9
402 0898 9
403 0899 9
404 0900 9
405 0901 9
406 0902 9
407 0903 9
408 0904 9
409 0905 9
410 0906 9
411 0907 9
412 0908 9
413 0909 9
414 0910 9
415 0911 9
416 0912 9
417 0913 10
418 0914 10
419 0915 10
420 0916 10
421 0917 10
422 0918 10
423 0919 10
424 0920 10
425 0921 10
426 0922 10
427 0923 9
428 0924 10
429 0925 10
430 0926 10
431 0927 11
432 0928 11
433 0929 11
434 0930 11
435 0931 11
436 0932 11
437 0933 11
438 0934 10
439 0935 9
440 0936 8
```



```
441      0937 7
442      0938 7
443      0939 6
444      0940 7
445      0941 7
446      0942 7
447      0943 7
448      0944 7
449      0945 7
450      0946 6
451      0947 5
452      0948 5
453      0949 5
454      0950 5
455      0951 5
456      0952 6
457      0953 6
458      0954 6
459      0955 6
460      0956 6
461      0957 6
462      0958 6
463      0959 6
464      0960 6
465      0961 6
466      0962 6
467      0963 6
468      0964 6
469      0965 7
470      0966 7
471      0967 7
472      0968 8
473      0969 8
474      0970 8
475      0971 8
476      0972 8
477      0973 9
478      0974 9
479      0975 9
480      0976 9
481      0977 9
482      0978 9
483      0979 9
484      0980 9
485      0981 8
486      0982 8
487      0983 8
488      0984 8
489      0985 8
490      0986 8
491      0987 7
492      0988 8
493      0989 8
494      0990 8
495      0991 8
496      0992 8
497      0993 9

      end ;
    else
      begin
        if .lnk$gl_curclu [clu$v_based] or .lnk$gl_curclu [clu$v_usrbased]
        then
          lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
            .curhdrisd [isd$w_pagcnt] * 512 ;
        end ;
      end ;
    end ;

    if not .curhdrisd [isd$v_gbl]
    then
      begin
        lnk$alloblk (isd$c_size, curisd) ;
        curisd [isl$nextisd] = 0 ;
        curisd [isl$bufadr] = 0 ;
        curisd [isl$bufend] = 0 ;
        curisd [isl$cludsc] = .lnk$gl_curclu ;
        newhdrisd = curisd [isl$hdrisd] ;
        ch$copy (.curhdrisd [isd$w_size],
          , curhdrisd [isd$w_size], 0
          , isd$k_maxlengthbl, curisd [isl$hdrisd]) ;
        if .lnk$gl_curclu [clu$l_fstisd] eql 0
        then
          begin
            if .lnk$gl_curclu [clu$v_usrbased]
            then
              begin
                if .lnk$gl_curclu [clu$v_based]
                and
                  .lnk$gl_curclu [clu$l_base] neq .lnk$gl_curclu [clu$l_usrbase]
                then
                  begin
                    signal ( lin$confbasadr, 5
                      , lnk$gl_curclu [clu$b_namlng]
                      , .lnk$gl_curclu [clu$l_base], lnk$gl_curfil [fdb$q_filename]
                      , .lnk$gl_curclu [clu$l_usrbase], $descriptor ('Options file')
                      , lin$noimgfil
                      ) ;
                    lnk$gl_ctlmsk [lnk$v_image] = false ;
                  end ;
                lnk$gl_curclu [clu$l_base] = .lnk$gl_curclu [clu$l_usrbase] ;
                lnk$gl_curclu [clu$l_usrbase] = 0 ;
                lnk$gl_curclu [clu$v_based] = true ;
              end
            else
              begin
                if .lnk$gl_curclu [clu$v_based]
                and not
                  .lnk$gl_curclu [clu$v_pic]
                then
                  begin
                    ! ALLOCATE A DESCRIPTOR
                    ! CURRENT IS LAST
                    ! NO BUFFER FOR IT YET
                    ! SET POINTER TO CLUSTER DESCRIPTOR
                    ! POINT TO PART DESTINED FOR HEADER
                    ! COPY THE ISD FROM THE
                    ! HEADER TO DESCRIPTOR 0 FILLED
                    ! IF THIS IS THE FIRST
                    ! IF BASED BY USER
                    ! IF CLUSTER IS ALSO BASED
                    ! IF BASED DUE TO ANOTHER IMAGE
                    ! BASING IT AND IT TURNED OUT TO
                    ! BE NON-PIC
                  end
                end
              end
            end
          end
        end
      end
    end
```

```

: 498      0994  9
: 499      0995  9
: 500      0996 10
: 501      0997 10
: 502      0998 10
: 503      0999 10
: 504      1000 10
: 505      1001 10
: 506      1002 10
: 507      1003 10
: 508      1004 10
: 509      1005 10
: 510      1006 10
: 511      1007  9
: 512      1008  8
: 513      1009  8
: 514      1010  8
: 515      1011  9
: 516      1012  9
: 517      1013  9
: 518      1014  8
: 519      1015  7
: 520      1016  7
: 521      1017  6
: 522      1018  6
: 523      1019  6
: 524      1020  6
: 525      1021  6
: 526      1022  7
: 527      1023  7
: 528      1024  7
: 529      1025  7
: 530      1026  7
: 531      1027  7
: 532      1028  6
: 533      1029  6
: 534      1030  6
: 535      1031  6
: 536      1032  6
: 537      1033  7
: 538      1034  7
: 539      1035  7
: 540      1036  7
: 541      1037  7
: 542      1038  7
: 543      1039  7
: 544      1040  7
: 545      1041  7
: 546      1042  7
: 547      1043  7
: 548      1044  6
: 549      1045  6
: 550      1046  6
: 551      1047  6
: 552      1048  6
: 553      1049  6
: 554      1050  6

      if .lnk$gl_curclu [clu$l_base] neq .newhdrisd [isd$v_vpg]^9
      then
      begin
        ownclu = .lnk$gl_curclu [clu$l_ownclu] ;
        ownfdb = .ownclu [clu$l_fstfdb] ;
        signal ( lin$ confbasadr, 5
                  , lnk$gl_curclu [clu$b_namlng]
                  , .newhdrisd [isd$v_vpg]^9, lnk$gl_curfil [fdb$q_filename]
                  , .lnk$gl_curclu [clu$l_base], ownfdb [fdb$q_filename]
                  , lin$_noimgfil
                ) ;
        lnk$gl_ctlmsk [lnk$v_image] = false ;
      end
    else
      if not .lnk$gl_curclu [clu$v_pic]
      then
        ! THEN EXTRACT BASE VPN
        begin
          lnk$gl_curclu [clu$l_base] = .newhdrisd [isd$v_vpg]^9 ;
          lnk$gl_curclu [clu$v_based] = true ;
        end ;
      end ;
      firstisd = .newhdrisd ;
      ! POINT TO FIRST ISECT IN CLUSTER
    end ;
    lnk$gl_curclu [clu$l_nisects] = .lnk$gl_curclu [clu$l_nisects] + 1 ;
    ! COUNT ISECT IN CLUSTER
    begin
      bind lastisd = lnk$gl_curclu [clu$l_lstisd]
          : ref block [, byte] ;
      ! POINTER TO LAST ISD IN CLUSTER
      lastisd [isl$l_nxtisd] = .curisd ;
      ! PUT AT END OF LIST
      lastisd = .curisd ;
      ! AND MAKE CURRENT THE NEW LAST
    end ;
    if .lnk$gl_curclu [clu$v_pic]
    then
      ! IF A PIC CLUSTER
      begin
        if .newhdrisd eql .firstisd
        then
          ! IF FIRST ISECT THIS CLUSTER
          firstisdvpg = .newhdrisd [isd$v_vpg] ;
          ! THEN SAVE VPG OF FIRST ISECT
          newhdrisd [isd$v_vpg] = .newhdrisd [isd$v_vpg] -
            .firstisdvpg ;
          ! THEN SUBTRACT OUT THE BASE
          ! FOR LATER RE-LOCATION (NEEDED
          ! FOR OLD IMAGES LINKED @200)
        if not .lnk$gl_curclu [clu$v_based]
        then lnk$gw_shriscts = .lnk$gw_shriscts + 1 ;
          ! COUNT IT
        end ;
      lnk$gl_curclu [clu$l_pages] = .lnk$gl_curclu [clu$l_pages] + .newhdrisd [isd$v_pagcnt] ;
      ! ACCUMULATE THE PAGES REQUIRED
    if (ownclu = .lnk$gl_curclu [clu$l_ownclu]) eql 0
    then
```



```
: 555      1051  6      lnk$gl_curclu [clu$l_cluoff] = .lnk$gl_curclu [clu$l_cluoff] +
: 556      1052  6      .newhdrisd [isd$w_pagcnt]*512
: 557      1053  6      else
: 558      1054  6      ownclu [clu$l_cluoff] = .ownclu [clu$l_cluoff] + .newhdrisd [isd$w_pagcnt]*512 ;
: 559      1055  6
: 560      1056  6      if .curhdrisd [isd$v_wrt]                                ! IF SECTION IS WRITEABLE
: 561      1057  6      and not
: 562      1058  6      .curhdrisd [isd$v_crf]                                ! AND NOT COPY-ON-REF
: 563      1059  6      then
: 564      1060  6      lnk$gl_curclu [clu$v_wrt] = true ;                    ! THEN REMEMBER FOR LNKIMGOUT
: 565      1061  6
: 566      1062  6      if .lnk$gl_curclu [clu$v_based]                        ! IF CLUSTER IS BASED THEN FLAG
: 567      1063  6      then
: 568      1064  6      newhdrisd [isd$v_based] = true ;                    ! IN CLUSTER DESCRIPTOR ALSO
: 569      1065  6
: 570      1066  6      if not .curhdrisd [isd$v_dzro]                        ! PROVIDED NOT A DEMAND ZERO ISD
: 571      1067  6      and .curhdrisd [isd$l_vbn] neq 0                    ! AND SECTION IS PRESENT IN IMAGE
: 572      1068  6      then
: 573      1069  6      if .curhdrisd [isd$l_vbn] gequ .maxisdvbn            ! IF IMAGE SECTION GOES BEYOND LAST
: 574      1070  6      then
: 575      1071  6      maxisdvbn = .curhdrisd [isd$l_vbn] +                ! BLOCK OF LAST ISECT, THEN
: 576      1072  6      .curhdrisd [isd$w_pagcnt] ;                        ! COMPUTE NEW MAX VBN IN USE
: 577      1073  6
: 578      1074  6      if not .curhdrisd [isd$v_gbl]                        ! IF IMAGE SECTION IS NOT
: 579      1075  6      then                                                ! ALREADY GLOBAL
: 580      1076  7      begin
: 581      1077  7      local gblsect_namlng ;
: 582      1078  7
: 583      1079  7      gblsect_namlng = .lnk$gl_curclu[clu$b_namlng] + suffix_size ;
: 584      1080  7
: 585      1081  7      The size of this global isd    = length of private isd
: 586      1082  7      + length of gblsect ident
: 587      1083  7      + length of gblsect name count byte
: 588      1084  7      + length of gblsect name
: 589      1085  7
: 590      1086  7      newhdrisd [isd$w_size] = isd$k_lenpriv + .gblsect_namlng + 5 ; ! SET SIZE AND
: 591      1087  7      newhdrisd [isd$v_gbl] = true ;                        ! MAKE IT GLOBAL NOW
: 592      1088  7      ! GLOBAL ISDS
: 593      1089  7      ! COMPUTE ISD NAME BY
: 594      1090  8      decr i from 3 to 1
: 595      1091  8      do begin
: 596      1092  8      if (curisdseq [.i] = .curisdseq [.i] + 1) gtru %c'9' ! INCREMENTING THE SUFFIX
: 597      1093  8      then
: 598      1094  8      curisdseq [.i] = %c'0'
: 599      1095  8      else
: 600      1096  8      exitloop ;
: 601      1097  7      end ;
: 602      1098  7
: 603      1099  7      (newhdrisd [isd$t_gblnam])<0, 8, 0> = .gblsect_namlng ;
: 604      1100  7
: 605      1101  7      ! COPY THE CLUSTER
: 606      1102  7      ! NAME CONCATENATED WITH
: 607      1103  7      ch$copy (.lnk$gl_curclu [clu$b_namlng]
: 608      1104  7      ,lnk$gl_curclu [clu$t_name], 4
: 609      1105  7      ,curisdseq [0], 0, .gblsect_namlng
: 610      1106  7      ,newhdrisd [isd$t_gblnam] + 1
: 611      1107  7      ) ;
:                                     ! FILL THEN SET MATCH CONTROL
:                                     newhdrisd [isd$v_matchctl] = .lnk$gl_curclu [clu$v_matchctl] ;
```

```
612      newhdrisd [isd$l_ident] = .isectident ;      ! AND THE MATCH CONTROL IDENT
613      end ;
614
615      if not .curhdrisd [isd$v_dzro]      ! IF NOT DEMAND ZERO AND
616      and .lnk$gl_curclu [clu$v_copy]      ! SHAREABLE IMAGE TO BE COPIED
617      and .curhdrisd [isd$l_vbn] neq 0      ! AND SECTION IS PRESENT IN THIS
618      then      ! ONE
619      begin
620      if .curhdrisd [isd$w_pagcnt] gtru .lnk$gl_gsbuofdsc [0]      ! MAXIMIZE THE SIZE OF
621      then      ! BUFFER WE'LL NEED
622      lnk$gl_gsbuofdsc [0] = .curhdrisd [isd$w_pagcnt] ;
623      end ;
624
625      if .iafva neq 0      ! IF THERE IS A FIXUP SECTION
626      and      !
627      (.newhdrisd [isd$v_vpg]^9 eql .iafva)      ! AND THIS IS IT
628      then
629      begin
630      lnk$gl_curclu [clu$l_fixisd] = .curisd ;      ! THEN REMEMBER IT FOR LATER
631      newhdrisd [isd$v_fixupvec] = true ;      ! FLAG FIXUP SECTION IN ISD
632      end ;      ! UNCONDITIONALLY
633
634      lnk$gw_nisects = .lnk$gw_nisects + 1 ;      ! COUNT THAT IMAGE SECTION
635      end ;      ! OF LOCAL ISECT
636
637      if (curhdrisd = .curhdrisd + .nxtisdoft) gtru (.header + 510) ! CHECK IT WAS COMPLETELY
638      then      ! CONTAINED BY THE CURRENT BLOCK
639      signal_stop (lin$_badimghdr, 1, lnk$gl_curfil [fdb$q_filename] ! IF NOT TRUE, FATAL IMAGE
640      ;      ! HEADER ERROR
641      ) ;
642      end      ! END OF ISECTION LOOP
643      else
644      :
645      : IMAGE SECTION WAS A STACK ISD, JUST SKIP IT BUT MAKE SURE IT IS CONTAINED BY THE
646      : CURRENT BLOCK -- ISSUE ERROR AND QUIT IF NOT
647      :
648      if (curhdrisd = .curhdrisd + .nxtisdoft) gtru (.header + 510)
649      then
650      signal_stop (lin$_badimghdr, 1, lnk$gl_curfil [fdb$q_filename]) ;
651      end ;
652      end ;      ! END OF BLOCK LOOP
653      end ;
654
655      if .maxisdvbn gtru .gstvbn [0] or .hdrblkcnt neq 0      ! AND THIS SHOULD ALSO POINT TO GST
656      then      ! AND NO HEADER BLOCKS REMAIN
657      signal_stop (lin$_badimghdr, 1, lnk$gl_curfil [fdb$q_filename]) ; ! IF EITHER ABOVE NOT TRUE,
658      ! FATAL IMAGE HEADER ERROR
659      end ;
660      if .lnk$gl_curclu [clu$v_based]      ! IF THIS CLUSTER IS BASED
661      and .lnk$gl_defclu [clu$v_based]      ! AND BASE= IN OPTION FILE
662      and .lnk$gl_curclu [clu$l_base] lequ .lnk$gl_defclu [clu$l_base] ! AND THIS IMAGE IS BELOW IT
663      then
664      signal (lin$_basshrbel, 3, lnk$gl_curfil [fdb$q_filename] ! THEN WARN USER SHR IMG
665      , lnk$gl_curclu [clu$l_base], .lnk$gl_defclu [clu$l_base] ! BELOW BASE=
666      ) ;
667
668      gstvbn [0] = .gstvbn [0] + .blockoffset ;      ! RELOCATE DOWN FILE IF NECESSARY
```



```

LNK$ALLOBLK, LNK$ALLOCLUSTER
LNK$CLOSEFILE, LNK$POINTOBJ
LNK$ADDIMAGE, LNK$PROCSOBJ
LNK$_BADIMGHDR, LNK$_BASSHRBEL

```

				OFFC 00000		
		5E	A8	AE 9E 00002		
				6C 95 00006		
				04 12 00008		
				59 D4 0000A		
				05 11 0000C		
59	04	BC		01 C3 0000E 1\$:		
			00000000G	00 D6 00013 2\$:		
			40	AE D4 00019		
	00000000'	EF		01 D0 0001C		
	1C	AE		01 CE 00023		
			24	AE D4 00027		
	00000000'	EF		59 D0 0002A		
18		00	00303030	8F F0 00031		
	2C	AE		A9 9E 0003E		
	FFFFFFFF	8F	1C	AE D1 00043 3\$:		
				03 13 0004B		
			0613	31 0004D		
	00000000V	EF		00 FB 00050 4\$:		
	17			50 E8 00057		
7E	00000000G	00		14 C1 0005A		
				01 DD 00062		
			00000000G	8F DD 00064		
	00000000G	00		03 FB 0006A		
	50	00000000G		00 D0 00071 5\$:		
	2C	AE	00000000'	EF D1 00078		
				03 13 00080		
			01C2	31 00082		
		02	11	A0 91 00085 6\$:		
				71 12 00089		
	3230	8F	0C	A0 B1 0008B		
				69 12 00091		
	30	0E	A0	91 00093		
			06	12 00097		
	35	0F	A0	91 00099		
			5D	1A 0009D		
	51	08	A0	3C 0009F 7\$:		
	51		2C	C0 000A3		
	000000A8	8F		51 D1 000A6		
				04 1E 000AD		
		51	A8	8F 9A 000AF		
51		60	10	00 ED 000B3 8\$:		

		.EXTRN	LIN\$	CONFASADR	
		.EXTRN	LIN\$	IDMISMCH, LIN\$	IMGBASED
		.EXTRN	LIN\$	NOIMGFIL, LIN\$	NONPICIMG
		.EXTRN	LIN\$	RELINK, LIN\$	READERR
		.EXTRN	LNK\$GL	SHRCLSTRS	
		.EXTRN	LNK\$AL	RAB, LNK\$GW	NISECTS
		.EXTRN	LNK\$GW	SHRISCTS	
		.EXTRN	LNK\$GB	PASS, LNK\$GL	CURFIL
		.EXTRN	LNK\$GL	CURCLU, LNK\$GL	DEFCLU
		.EXTRN	LNK\$GL	LASTCLU, LNK\$GL	CTLMSK
		.EXTRN	LNK\$GL	OBJRECS	
		.PSECT	\$CODE\$	NOWRT, 2	
		.ENTRY	LNK\$PROC	SHRIM, Save R2,R3,R4,R5,R6,R7,R8,-	0652
				R9,R10,R11	
		MOVAB		-88(SP), SP	
		TSTB		(AP)	0715
		BNEQ		1\$	
		CLRL		BLOCKOFFSET	0717
		BRB		2\$	
		SUBL3		#1, @MODRFA, BLOCKOFFSET	0719
		INCL		LNK\$GL SHRCLSTRS	0721
		CLRL		CLUDEST	0723
		MOVL		#1, HDRBLKCNT	0724
		MNEGL		#1, NXTISDOFF	0725
		CLRL		MAXISDVBN	0726
		MOVL		BLOCKOFFSET, HEADERBLOCK	0727
		INSV		#3158064, #0, #24, CURISDSEQ+1	0729
		MOVAB		1(R9), 44(SP)	0738
		CMPL		NXTISDOFF, #-1	0731
		BEQL		4\$	
		BRW		59\$	
		CALLS		#0, READNEXTBLOCK	0733
		BLBS		R0, 5\$	
		ADDL3		#20, LNK\$GL_CURFIL, -(SP)	0736
		PUSHL		#1	
		PUSHL		#LIN\$ BADIMGHDR	
		CALLS		#3, LTB\$STOP	
		MOVL		HEADER, R0	0744
		CMPL		HEADERBLOCK, 44(SP)	0738
		BEQL		6\$	
		BRW		17\$	
		CMPB		17(R0), #2	0744
		BNEQ		9\$	
		CMPW		12(R0), #12848	0745
		BNEQ		9\$	
		CMPB		14(R0), #48	0746
		BNEQ		7\$	
		CMPB		15(R0), #53	0748
		BGTRU		9\$	
		MOVZWL		8(R0), R1	0750
		ADDL2		#44, R1	
		CMPL		R1, #168	
		BGEQU		8\$	
		MOVZBL		#168, R1	
		CMPZV		#0, #16, (R0), R1	

				42	1A	000B8	BGTRU	9\$		
				A0	9A	000BA	MOVZBL	16(R0), HDRBLKCNT	0754	
				EF	D7	000C2	DECL	HDRBLKCNT		
				32	19	000C8	BLSS	9\$		
				A0	3C	000CA	MOVZWL	4(R0), SYMDBGDATDSC	0755	
				50	C0	000CE	ADDL2	R0, SYMDBGDATDSC		
				60	3C	000D1	MOVZWL	(R0), R1	0756	
				51	C0	000D4	ADDL2	R1, R0		
				58	D1	000D7	CMPL	SYMDBGDATDSC, R0		
				20	1E	000DA	BGEQU	9\$		
				A8	3C	000DC	MOVZWL	10(SYMDBGDATDSC), GSTRECS	0757	
				AE	D1	000E1	CMPL	GSTRECS, #3		
				15	1F	000E5	BLSSU	9\$		
				A8	D0	000E7	MOVL	4(SYMDBGDATDSC), R0	0758	
				50	D0	000EB	MOVL	R0, GSTVBN		
				01	C1	000EF	ADDL3	#1, HDRBLKCNT, R1	0759	
				50	D1	000F7	CMPL	R0, R1		
				17	1A	000FA	BGTRU	10\$		
				14	C1	000FC	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	0762	
				01	DD	00104	PUSHL	#1		
				8F	DD	00106	PUSHL	#LINS_BADIMGHDR		
				03	FB	0010C	CALLS	#3, LIB\$STOP		
				00	D0	00113	MOVL	LNK\$GL_CURCLU, R0	0764	
				00	D0	0011A	MOVL	HEADER, R1		
				03	EF	00121	EXTZV	#3, #1, 32(R1), R1		
				51	F0	00127	INSV	R1, #3, #1, 88(R0)		
				51	E8	0012D	BLBS	R1, 11\$		
				02	8A	00130	BICB2	#2, LNK\$GL_CTLMSK+2	0767	
				02	E1	00137	BBC	#2, 89(R0), 12\$	0769	
				01	E0	0013C	BBS	#1, 89(R0), 12\$	0770	
				14	C1	00141	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	0774	
				01	DD	00149	PUSHL	#1		
				8F	DD	0014B	PUSHL	#LINS_IMGBASED		
				03	FB	00151	CALLS	#3, LIB\$SIGNAL		
				00	D0	00158	MOVL	LNK\$GL_CURCLU, R0	0776	
				04	8A	0015F	BICB2	#4, 89(R0)		
				A0	D4	00163	CLRL	60(R0)	0777	
				0C	11	00166	BRB	12\$	0764	
				02	E1	00168	BBC	#2, 89(R0), 12\$	0781	
				02	8A	0016D	BICB2	#2, LNK\$GL_CTLMSK+2	0783	
				02	E1	00174	BBC	#2, LNK\$GL_CTLMSK, 13\$	0785	
				03	E0	0017C	BBS	#3, LNK\$GL_CTLMSK+3, 13\$	0786	
				00	D0	00184	MOVL	LNK\$GL_CURCLU, R0	0787	
				03	E0	0018B	BBS	#3, 88(R0), 13\$		
				14	C1	00190	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	0790	
				01	DD	00198	PUSHL	#1		
				8F	DD	0019A	PUSHL	#LINS_NONPICIMG		
				03	FB	001A0	CALLS	#3, LIB\$SIGNAL		
				00	D0	001A7	MOVL	LNK\$GL_CURCLU, R1	0792	
				00	D0	001AE	MOVL	HEADER, R0	0793	
				00	EF	001B5	EXTZV	#0, #3, 35(R0), R2		
				52	F0	001BB	INSV	R2, #5, #3, 89(R1)		
				52	D1	001C1	CMPL	R2, #3		
				04	12	001C4	BNEQ	14\$		
				10	88	001C6	BISB2	#16, 88(R1)	0795	
				A0	D0	001CA	MOVL	36(R0), ISECTIDENT	0797	
				AE	D4	001CF	CLRL	IAFVA	0798	

		52	02	A0	3C	001D2	MOVZWL	2(R0), R2	0800	
		52		50	C0	001D6	ADDL2	R0, R2		
		53	2C	A0	9E	001D9	MOVAB	44(R0), R3	0801	
		53		52	D1	001DD	CMPL	R2, R3		
				07	1B	001E0	BLEQU	15\$		
	20	AE	2C	A0	D0	001E2	MOVL	44(R0), IAFVA	0804	
				04	12	001E7	BNEQ	16\$	0805	
	59	A1		01	88	001E9	BISB2	#1, 89(R1)	0810	
52	00000000	EF	00000000	EF	C1	001ED	ADDL3	HDRBLKCNT, LNK\$GL_IMGRECS, R2	0812	
	00000000	EF	01	A2	9E	001F9	MOVAB	1(R2), LNK\$GL_IMGRECS		
		56		60	3C	00201	MOVZWL	(R0), CURHDRISD	0813	
		56		50	C0	00204	ADDL2	R0, CURHDRISD		
		51	0084	C1	9E	00207	MOVAB	132(R1), R1	0819	
				61	D5	0020C	TSTL	(R1)		
				3A	13	0020E	BEQL	18\$		
	38	AE		61	D1	00210	CMPL	(R1), ISECTIDENT	0820	
				34	13	00214	BEQL	18\$		
		50	00000000G	00	D0	00216	MOVL	LNK\$GL_CURFIL, R0	0829	
			1C	A0	9F	0021D	PUSHAB	28(R0)		
7E		61		00	EF	00220	EXTZV	#0, #24, (R1), -(SP)		
		7E		A1	9A	00225	MOVZBL	3(R1), -(SP)		
				A0	9F	00229	PUSHAB	20(R0)	0828	
7E	48	AE		00	EF	0022C	EXTZV	#0, #24, IMGIDENT, -(SP)	0829	
				7E	4F	AE	9A	00232		
					06	DD	00236	PUSHL	#6	
					8F	DD	00238	PUSHL	#LINS_IDMISMCH	
	00000000G	00	00000000G	08	FB	0023E	CALLS	#8, LIB\$SIGNAL		
				03	11	00245	BRB	18\$	0738	
		56		50	D0	00247	MOVL	R0, CURHDRISD	0834	
	1C	AE		66	32	0024A	CVTBL	(CURHDRISD), NXTISDOFF	0839	
				03	14	0024E	BGTR	19\$		
				FD	8F	0B	A6	91 00253	19\$:	
					03	12	00258	BNEQ	20\$	
					03D7	31	0025A	BRW	57\$	
	14	AE	08	A6	9E	0025D	MOVAB	8(CURHDRISD), 20(SP)	0844	
				08	E1	00262	BBC	#8, @20(SP), 21\$		
1B	14	BE		14	C1	00267	ADDL3	#20, LNK\$GL_CURFIL, R0	0848	
50	00000000G	00		50	DD	0026F	PUSHL	R0		
				50	DD	00271	PUSHL	R0		
				02	DD	00273	PUSHL	#2		
	00000000G	00	00000000G	8F	DD	00275	PUSHL	#LINS_RELINK		
		03		04	FB	0027B	CALLS	#4, LIB\$STOP		
			14	BE	E8	00282	BLBS	@20(SP), 22\$	0851	
				0149	31	00286	BRW	37\$		
		50	14	A6	9E	00289	MOVAB	20(CURHDRISD), R0	0859	
				01	A0	9E	0028D	MOVAB	1(R0), SHRDESC+4	
	4C	AE	01	A6	9B	00292	MOVZBW	20(CURHDRISD), SHRDESC	0860	
	48	AE	14	04	A2	00297	SUBW2	#4, SHRDESC		
	48	AE		AE	D0	0029B	MOVL	CLUDESC, R1	0862	
		51	40	0C	13	0029F	BEQL	23\$		
5D	A1	01	A0	48	AE	29 002A1	CMPC3	SHRDESC, 1(R0), 93(R1)	0864	
					03	12	002A8	BNEQ	23\$	
					0102	31	002AA	BRW	34\$	
				3C	AE	9F	002AD	PUSHAB	FOUND	
				44	AE	9F	002B0	PUSHAB	CLUDESC	
					7E	D4	002B3	CLRL	-(SP)	

59	A2	01	00	54	AE	9F	002B5	PUSHAB	SHRDESC		
		03	58	54	04	FB	002B8	CALLS	#4, LNK\$ADDIMAGE		0872
		04	14	58	00	DO	002BF	MOVL	LNK\$GL_CURCLU, R0		
		04	53	40	A0	9E	002C6	MOVAB	88(R0), R1		0869
		04	53	3C	AE	DO	002CA	MOVL	CLUDESC, R2		
		04	53	01	38	13	002CE	BEQL	28\$		
		04	53	01	AE	E8	002D0	BLBS	FOUND, 28\$		
		04	53	01	A1	F0	002D4	INSV	1(R1), #0, #1, 89(R2)		0872
		04	53	01	50	DO	002DB	MOVL	R0, 84(R2)		0874
		04	53	01	03	E1	002DF	BBC	#3, (R1), 25\$		0876
		04	53	01	00E5	31	002E3	BRW	36\$		
		04	53	01	01	88	002E6	BISB2	#1, 88(R2)		0879
		04	53	01	09	E1	002EA	BBC	#9, 20(SP), 26\$		0880
		04	53	01	00	EF	002EF	EXTZV	#0, #23, 4(CURHDRISD), R3		0881
		04	53	01	09	78	002F5	ASHL	#9, R3, R3		
		04	53	01	06	11	002F9	BRB	27\$		
		04	53	01	A0	C1	002FB	ADDL3	32(R0), 76(R0), R3		0883
		04	53	01	53	DO	00301	MOVL	R3, 76(R2)		0880
		04	53	01	00B7	31	00305	BRW	35\$		0886
		04	53	01	61	E9	00308	BLBC	(R1), 29\$		0891
		04	53	01	03	E1	0030B	BBC	#3, (R1), 30\$		0893
		04	53	01	0A	E1	0030F	BBC	#10, (R1), 24\$		0895
		04	53	01	A2	DO	00313	MOVL	84(R2), OWNCLU		0898
		04	53	01	54	D4	00317	CLRL	R4		0900
		04	53	01	6E	D5	00319	TSTL	OWNCLU		
		04	53	01	0E	13	0031B	BEQL	31\$		
		04	53	01	54	D6	0031D	INCL	R4		
		04	53	01	A6	3C	0031F	MOVZWL	2(CURHDRISD), R3		0903
		04	53	01	09	78	00323	ASHL	#9, R3, R3		
		04	53	01	53	C0	00327	ADDL2	R3, 32(R0)		
		04	53	01	54	E9	0032B	BLBC	R4, 32\$		0905
		04	53	01	8F	C1	0032E	ADDL3	#88, OWNCLU, R0		0907
		04	53	01	60	E9	00336	BLBC	(R0), 32\$		
		04	53	01	00	EF	00339	EXTZV	#0, #23, 4(CURHDRISD), R3		0909
		04	53	01	09	78	0033F	ASHL	#9, R3, R3		
		04	53	01	A2	D1	00343	CMPL	76(R2), R3		
		04	53	01	45	13	00347	BEQL	32\$		
		04	53	01	08	C1	00349	ADDL3	#8, OWNCLU, R0		0914
		04	53	01	60	DO	0034D	MOVL	(R0), OWNFDB		
		04	53	01	8F	DD	00351	PUSHL	#LINS NOIMGFIL		0918
		04	53	01	14	C1	00357	ADDL3	#20, OWNFDB, R0		
		04	53	01	50	DD	0035C	PUSHL	R0		
		04	53	01	A2	DD	0035E	PUSHL	76(R2)		
		04	53	01	14	C1	00361	ADDL3	#20, LNK\$GL_CURFIL, -(SP)		0917
		04	53	01	00	EF	00369	EXTZV	#0, #23, 4(CURHDRISD), R0		
		04	53	01	09	78	0036F	ASHL	#9, R0, -(SP)		
		04	53	01	A2	9F	00373	PUSHAB	92(R2)		0916
		04	53	01	05	DD	00376	PUSHL	#5		0918
		04	53	01	8F	DD	00378	PUSHL	#LINS CONFBASADR		
		04	53	01	08	FB	0037E	CALLS	#8, LIB\$SIGNAL		
		04	53	01	01	8A	00385	BICB2	#1, LNK\$GL_CTLMSK		0921
		04	53	01	3D	11	0038C	BRB	36\$		0905
		04	53	01	0A	E0	0038E	BBS	#10, (R1), 36\$		0925
		04	53	01	A2	E8	00392	BLBS	88(R2), 33\$		0928
		04	53	01	A2	A2	00396	SUBW2	72(R2), LNK\$GW_SHRISCTS		0930
		04	53	01	01	88	0039E	BISB2	#1, 88(R2)		0932
		04	53	01	00	EF	003A2	EXTZV	#0, #23, 4(CURHDRISD), R0		0933

4C	A2	50	09	78	003A8	ASHL	#9, R0, 76(R2)	0891	
		50	1C	11	003AD	BRB	36\$	0941	
		05	00	D0	003AF	MOVL	LNK\$GL_CURCLU, R0		
		58	A0	E8	003B6	BLBS	88(R0), 35\$		
0C	59	A0	02	E1	003BA	BBC	#2, 89(R0), 36\$		
		53	02	A6	3C	003BF	35\$: MOVZWL	0944	
53		53	09	78	003C3	ASHL	#9, R3, R3		
	20	A0	53	C0	003C7	ADDL2	R3, 32(R0)		
		03	14	BE	E9	003CB	36\$: BLBC	0950	
			0262	31	003CF	BRW	57\$		
		7E	44	AE	9F	003D2	37\$: PUSHAB	0953	
		00	58	8F	9A	003D5	MOVZBL	#88, -(SP)	
00000000G		5B	44	02	FB	003D9	CALLS	#2, LNK\$ALLOBLK	
			08	AB	D0	003E0	MOVL	CURISD, R11	0954
		5A	00	6B	D4	003E4	CLRL	(R11)	
	10	AB	00	7C	003E6	CLRQ	8(R11)	0955	
		57	5A	D0	003E9	MOVL	LNK\$GL_CURCLU, R10	0957	
0040	8F	66	18	AB	9E	003F0	MOVL	R10, 18(R11)	
			66	2C	003F8	MOVAB	24(R11), NEWHDRISD	0958	
			18	AB	003FF	MOVCS	(CURHDRISD), (CURHDRISD), #0, #64, 24(R11)	0961	
			18	AA	D5	00401	TSTL	24(R10)	0963
			03	13	00404	BEQL	38\$		
			00C8	31	00406	BRW	43\$		
		51	58	AA	9E	00409	38\$: MOVAB	88(R10), R1	0966
52		61	0A	E1	0040D	BBC	#10, (R1), 40\$		
		3A	61	E9	00411	BLBC	(R1), 39\$	0969	
	3C	AA	4C	AA	D1	00414	CMP	76(R10), 60(R10)	0971
				33	13	00419	BEQL	39\$	
		00000000G	8F	DD	0041B	PUSHL	#LINS_NOIMGFIL	0976	
		00000000G	EF	9F	00421	PUSHAB	P.AAA	0977	
			3C	AA	DD	00427	PUSHL	60(R10)	
7E	00000000G	00	14	C1	0042A	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	0976	
			4C	AA	DD	00432	PUSHL	76(R10)	
			5C	AA	9F	00435	PUSHAB	92(R10)	0975
			05	DD	00438	PUSHL	#5	0976	
		00000000G	8F	DD	0043A	PUSHL	#LINS_CONFBASADR		
	00000000G	00	08	FB	00440	CALLS	#8, LIB\$SIGNAL		
		00	01	8A	00447	BICB2	#1, LNK\$GL_CTLMSK	0980	
		50	00	D0	0044E	39\$: MOVL	LNK\$GL_CURCLU, R0	0983	
	4C	A0	3C	A0	D0	00455	MOVL	60(R0), 76(R0)	
			3C	A0	D4	0045A	CLRL	60(R0)	0984
	58	A0	01	88	0045D	BISB2	#1, 88(R0)	0985	
			6A	11	00461	BRB	42\$	0966	
		55	61	E9	00463	40\$: BLBC	(R1), 41\$	0989	
		61	03	E0	00466	BBS	#3, (R1), 42\$	0991	
50		17	00	EF	0046A	EXTZV	#0, #23, 4(NEWHDRISD), R0	0994	
	04	50	09	78	00470	ASHL	#9, R0, R0		
		50	4C	AA	D1	00474	CMP	76(R10), R0	
			53	13	00478	BEQL	42\$		
		6E	54	AA	D0	0047A	MOVL	84(R10), OWNCLU	0997
	52	6E	08	C1	0047E	ADDL3	#8, OWNCLU, R2	0998	
		AE	62	D0	00482	MOVL	(R2), OWNFDB		
			8F	DD	00486	PUSHL	#LINS_NOIMGFIL	1002	
51		1C	AE	14	C1	0048C	ADDL3	#20, OWNFDB, R1	
				51	DD	00491	PUSHL	R1	
			4C	AA	DD	00493	PUSHL	76(R10)	

7E	00000000G	00	14	C1	00496	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	1001
			50	DD	0049E	PUSHL	R0	1002
		5C	AA	9F	004A0	PUSHAB	92(R10)	1000
		00000000G	05	DD	004A3	PUSHL	#5	1002
	00000000G	00	8F	DD	004A5	PUSHL	#LINS, CONFBASADR	
	00000000G	00	08	FB	004AB	CALLS	#8, LIB\$SIGNAL	
			01	8A	004B2	BICB2	#1, LNK\$GL_CTLMSK	1005
			12	11	004B9	BRB	42\$	0993
	OE	61	03	E0	004BB	BBS	#3, (R1), 42\$	1009
50	A7	17	00	EF	004BF	EXTZV	#0, #23, 4(NEWHDRISD), R0	1012
04	AA	50	09	78	004C5	ASHL	#9, R0, 76(R10)	
		61	01	88	004CA	BISB2	#1, (R1)	1013
		AE	57	D0	004CD	MOVL	NEWHDRISD, FIRSTISD	1016
		5A	00	D0	004D1	MOVL	LNK\$GL_CURCLU, R10	1019
			48	AA	D6	INCL	72(R10)	
	1C	BA	5B	D0	004DB	MOVL	R11, @28(R10)	1026
	1C	AA	5B	D0	004DF	MOVL	R11, 28(R10)	1027
	04	AE	58	AA	9E	MOVAB	88(R10), 4(SP)	1031
	27	BE	03	E1	004E3	BBC	#3, @4(SP), 45\$	
	04	AE	57	D1	004ED	CMPL	NEWHDRISD, FIRSTISD	1034
	30		07	12	004F1	BNEQ	44\$	
34	AE	17	00	EF	004F3	EXTZV	#0, #23, 4(NEWHDRISD), FIRSTISDVPG	1036
50	A7	17	00	EF	004FA	EXTZV	#0, #23, 4(NEWHDRISD), R0	1039
		50	AE	C2	00500	SUBL2	FIRSTISDVPG, R0	
04	A7	00	50	F0	00504	INSV	R0, #0, #23, 4(NEWHDRISD)	
		06	BE	E8	0050A	BLBS	@4(SP), 45\$	1042
			00	B6	0050E	INCW	LNK\$GW_SHRISCTS	1043
		50	02	A7	3C	MOVZWL	2(NEWHDRISD), R0	1046
		AA	50	C0	00518	ADDL2	R0, 80(R10)	
		6E	54	AA	D0	MOVL	84(R10), OWNCLU	1049
			0E	12	00520	BNEQ	46\$	
		50	02	A7	3C	MOVZWL	2(NEWHDRISD), R0	1052
		50	09	78	00526	ASHL	#9, R0, R0	
		20	50	C0	0052A	ADDL2	R0, 32(R10)	
			0F	11	0052E	BRB	47\$	1051
		50	02	A7	3C	MOVZWL	2(NEWHDRISD), R0	1054
		50	09	78	00534	ASHL	#9, R0, R0	
		51	20	C1	00538	ADDL3	#32, OWNCLU, R1	
		6E	50	C0	0053C	ADDL2	R0, (R1)	
	0A	14	03	E1	0053F	BBC	#3, @20(SP), 48\$	1056
	05	14	01	E0	00544	BBS	#1, @20(SP), 48\$	1058
		04	8F	88	00549	BISB2	#64, @4(SP)	1060
		04	BE	E9	0054E	BLBC	@4(SP), 49\$	1062
		09	02	88	00552	BISB2	#2, 9(NEWHDRISD)	1064
	16	14	02	E0	00556	BBS	#2, @20(SP), 50\$	1066
			0C	A6	D5	TSTL	12(CURHDRISD)	1067
			11	13	0055E	BEQL	50\$	
		24	0C	A6	D1	CMPL	12(CURHDRISD), MAXISDVBN	1069
			0A	1F	00565	BLSSU	50\$	
		50	02	A6	3C	MOVZWL	2(CURHDRISD), R0	1072
		24	0C	B640	9E	MOVAB	@12(CURHDRISD)[R0], MAXISDVBN	
		79	14	BE	E8	BLBS	@20(SP), 54\$	1074
		50	5C	AA	9A	MOVZBL	92(R10), GBLSECT_NAMLNG	1079
		50	04	C0	00579	ADDL2	#4, GBLSECT_NAMLNG	
67		50	15	A1	0057C	ADDW3	#21, GBLSECT_NAMLNG, (NEWHDRISD)	1086
		08	01	88	00580	BISB2	#1, 8(NEWHDRISD)	1087
			03	D0	00584	MOVL	#3, I	1089

			52	00000000'	EF41	9A	00587	51\$:	MOVZBL	CURISDSEQ[I], R2	1091			
						52	D6	0058F	INCL	R2				
				00000000'	EF41	52	90	00591	MOVB	R2, CURISDSEQ[I]				
					39	52	D1	00599	CMPL	R2, #57				
						0B	1B	0059C	BLEQU	52\$				
				00000000'	EF41	30	90	0059E	MOVB	#48, CURISDSEQ[I]	1093			
					DE	51	F5	005A6	SOBCTR	1, 51\$	1089			
				14	A7	50	90	005A9	52\$:	MOVB	GBLSECT_NAMLANG, 20(NEWHDRISD)	1098		
				10	AE	5C	AA	005AD	MOVZBL	92(R10), 16(SP)	1102			
				0C	AE	50	D0	005B2	MOVL	GBLSECT_NAMLANG, 12(SP)	1104			
				08	AE	15	A7	005B6	MOVAB	21(R7), 8(SP)	1105			
OC	AE		00	5D	AA	10	AE	2C	MOVCS	16(SP), 93(R10), #0, 12(SP), a8(SP)				
						08	BE	005C3						
				08	AE	16	18	005C5	BGEQ	53\$				
				0C	AE	10	AE	C0	ADDL2	16(SP), 8(SP)				
OC	AE		00	00000000'	EF	10	AE	C2	SUBL2	16(SP), 12(SP)				
						08	BE	005CC	MOVCS	#4, CURISDSEQ, #0, 12(SP), a8(SP)				
							04	2C						
							0D	EF	005DD	53\$:	EXTZV	#13, #3, a4(SP), R0	1107	
							50	F0	005E3		INSV	R0, #4, #3, 8(NEWHDRISD)		
							AE	D0	005E9		MOVL	ISECTIDENT, 16(NEWHDRISD)	1108	
				10	A7	38	AE	D0	005EE	54\$:	BBS	#2, a20(SP), 55\$	1111	
				1E	14		02	E0	005EE		BBC	#4, a4(SP), 55\$	1112	
				19	04	BE	04	E1	005F3		TSTL	12(CURHDRISD)	1113	
							0C	A6	D5	005F8	BEQL	55\$		
								14	13	005FB	CMPZV	#0, #16, 2(CURHDRISD), LNK\$GL_GSBUFDC	1116	
00000000'	EF		02	A6	10		00	ED	005FD		BLEQU	55\$		
							08	1B	00607		MOVZWL	2(CURHDRISD), LNK\$GL_GSBUFDC	1118	
				00000000'	EF		02	A6	3C	00609		TSTL	IAFVA	1121
							20	AE	D5	00611	55\$:	BEQL	56\$	
								18	13	00614		EXTZV	#0, #23, 4(NEWHDRISD), R0	1123
								00	EF	00616		ASHL	#9, R0, R0	
								09	78	0061C		CMPL	R0, IAFVA	
								50	D1	00620		BNEQ	56\$	
								08	12	00624		MOVL	R11, 68(R10)	1126
								5B	D0	00626		BISB2	#4, 9(NEWHDRISD)	1127
								04	88	0062A		INCW	LNK\$GL_NISECTS	1130
								00	B6	0062E	56\$:	ADDL2	NXTISDOFF, CURHDRISD	1144
								00	C0	00634	57\$:	ADDL3	#510, HEADER, R0	
								8F	C1	00638		CMPL	CURHDRISD, R0	
								56	D1	00644		BLEQU	58\$	
								17	1B	00647		ADDL3	#20, LNK\$GL_CURFIL, -(SP)	1146
								14	C1	00649		PUSHL	#1	
								01	DD	00651		PUSHL	#LINS_BADIMGHDR	
								8F	DD	00653		CALLS	#3, LIB\$STOP	
								03	FB	00659		BRW	18\$	0839
								31	00660	58\$:	CMPL	MAXISDVBN, GSTVBN	1151	
								AE	D1	00663	59\$:	BGTRU	60\$	
								08	1A	00668		TSTL	HDRBLKCNT	
								EF	D5	0066A		BEQL	61\$	
								17	13	00670		ADDL3	#20, LNK\$GL_CURFIL, -(SP)	1154
								14	C1	00672	60\$:	PUSHL	#1	
								01	DD	0067A		PUSHL	#LINS_BADIMGHDR	
								8F	DD	0067C		CALLS	#3, LIB\$STOP	
								03	FB	00682		MOVL	LNK\$GL_CURCLU, R0	1156
								00	D0	00689	61\$:	BLBC	88(R0), 62\$	
								A0	E9	00690		BLBC	LNK\$GL_DEFCLU+88, 62\$	1157
								E9	00694					

LNK_PROCSHRIM
V04=000

H 9
16-Sep-1984 00:30:18
14-Sep-1984 12:40:34

VAX-11 Bliss-32 V4.0-742
[LINKER.SRC]LNKPROSHR.B32;2

Page 21
(2)

00000000G	00	4C	A0	D1	0069B	CMPL	76(R0), LNK\$GL_DEFCLU+76	: 1158
			20	1A	006A3	BGTRU	62\$:
		00000000G	00	DD	006A5	PUSHL	LNK\$GL_DEFCLU+76	: 1161
		4C	A0	DD	006AB	PUSHL	76(R0)	:
7E	00000000G	00	14	C1	006AE	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	: 1160
			03	DD	006B6	PUSHL	#3	:
		00000000G	8F	DD	006B8	PUSHL	#LINS_BASSHRBEL	:
			05	FB	006BE	CALLS	#5, LIB\$SIGNAL	:
	50	AE	59	C0	006C5	ADDL2	BLOCKOFFSET, GSTVBN	: 1164
		54	AE	D4	006C9	CLRL	GSTVBN+4	: 1165
	52	00000000G	00	D0	006CC	MOVL	LNK\$GL_OBJRECS, SAVERECOUNT	: 1166
		50	AE	9F	006D3	PUSHAB	GSTVBN	: 1167
	00000000G	00	01	FB	006D6	CALLS	#1, LNK\$POINTOBJ	:
		50	AE	9F	006DD	PUSHAB	GSTVBN	: 1169
	00000000G	00	01	FB	006E0	CALLS	#1, LNK\$PROCSOBJ	:
	30		50	E9	006E7	BLBC	R0, 64\$:
52	00000000G	00	52	C3	006EA	SUBL3	SAVERECOUNT, LNK\$GL_OBJRECS, SAVERECOUNT	: 1173
	28	AE	52	D1	006F2	CMPL	SAVERECOUNT, GSTRECS	: 1175
			17	13	006F6	BEQL	63\$:
7E	00000000G	00	14	C1	006F8	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	: 1177
			01	DD	00700	PUSHL	#1	:
		00000000G	8F	DD	00702	PUSHL	#LINS_BADIMGHDR	:
			03	FB	00708	CALLS	#3, LIB\$STOP	:
	00000000G	00	52	C0	0070F	ADDL2	SAVERECOUNT, LNK\$GL_IMGRECS	: 1179
	00000000'	EF	01	D0	00716	MOVL	#1, R0	: 1180
		50		04	00719	RET		:
			50	D4	0071A	CLRL	R0	: 1181
			04	0071C	RET			:

; Routine Size: 1821 bytes, Routine Base: \$CODE\$ + 0000

; 686 1182 1

```

: 688      1183 1 routine readnextblock =                ! READ NEXT BLOCK IF ANY
: 689      1184 1 ++
: 690      1185 1      THIS ROUTINE IS CALLED TO READ THE NEXT BLOCK OF THE IMAGE HEADER
: 691      1186 1      READ ERRORS ARE FATAL. A REQUEST TO READ ANOTHER BLOCK
: 692      1187 1      WHEN HDRBLKCNT IS ALREADY ZERO RETURNS FATAL.
: 693      1188 1      HDRBLKCNT IS DECREMENTED AFTER EACH READ AND HEADERBLOCK IS
: 694      1189 1      INCREMENTED BEFORE EACH READ.
: 695      1190 1
: 696      1191 1 --
: 697      1192 2 begin
: 698      1193 2 local
: 699      1194 2     saveusz,
: 700      1195 2     readerror ;
: 701      1196 2
: 702      1197 2 if .hdrblkcnt leq 0                        ! IF NO MORE BLOCKS
: 703      1198 2 then
: 704      1199 2     return false ;                       ! RETURN FAILURE
: 705      1200 2
: 706      1201 2 saveusz = .lnk$al_rab [rab$w_usz] ;       ! SAVE USZ
: 707      1202 2 headerblock = .headerblock + 1 ;        ! SET THE BLOCK TO READ
: 708      1203 2
: 709      1204 2 lnk$al_rab [rab$l_bkt] = .headerblock ;   ! SET STARTING VBN
: 710      1205 2 lnk$al_rab [rab$w_usz] = 512 ;          ! AND SET THE BYTE COUNT
: 711      1206 2
: 712      1207 2 if not (readerror = $read (rab = lnk$al_rab)) ! ATTEMPT TO READ LIBRARY, USING
: 713      1208 2 then
: 714      1209 2     begin
: 715      1210 2     signal (lnk$readerr, 1
: 716      1211 2     ,lnk$gl_curfil [fdb$q_filename]
: 717      1212 2     ,readerror, .lnk$al_rab [rab$l_stv]
: 718      1213 2     ) ;
: 719      1214 2     lnk$closefile (.lnk$gl_curfil) ;      ! THE MESSAGES AND ATTEMPT TO
: 720      1215 2     lnk$al_rab [rab$w_usz] = .saveusz ;  ! RESTORE USZ
: 721      1216 2     return false ;
: 722      1217 2 end ;
: 723      1218 2
: 724      1219 2 hdrblkcnt = .hdrblkcnt - 1 ;            ! DECREMENT THE BLOCK COUNT
: 725      1220 2 lnk$al_rab [rab$w_usz] = .saveusz ;
: 726      1221 2 return true ;                            ! AND ALL DONE SUCCESSFULLY
: 727      1222 1 end ;                                    ! END OF READNEXTBLOCK

```

.EXTRN SYS\$READ

003C 00000 READNEXTBLOCK:						
55	00000000G	00	9E 00002	.WORD	Save R2,R3,R4,R5	: 1183
54	00000000	EF	9E 00009	MOVAB	LNK\$GL_CURFIL, R5	:
53	00000000G	00	9E 00010	MOVAB	HDRBLKCNT, R4	:
		64	D5 00017	MOVAB	LNK\$AL_RAB+32, R3	:
		4C	15 00019	TSTL	HDRBLKCNT	: 1197
52		63	3C 0001B	BLEQ	2\$:
	04	A4	D6 0001E	MOVZWL	LNK\$AL_RAB+32, SAVEUSZ	: 1201
18	A3	04	A4 D0 00021	INCL	HEADERBLOCK	: 1202
	63	0200	8F B0 00026	MOVL	HEADERBLOCK, LNK\$AL_RAB+56	: 1204
		E0	A3 9F 0002B	MOVW	#512, LNK\$AL_RAB+32	: 1205
				PUSHAB	LNK\$AL_RAB	: 1207

00000000G	00	01	FB	0002E	CALLS	#1, SYSSREAD	:
	26	50	EB	00035	BLBS	READERROR, 1\$:
		A3	DD	00038	PUSHL	LNK\$AL_RAB+12	1212
7E	65	50	DD	0003B	PUSHL	READERROR	:
		14	C1	0003D	ADDL3	#20, LNK\$GL_CURFIL, -(SP)	1211
		01	DD	00041	PUSHL	#1	:
		8F	DD	00043	PUSHL	#LINS_READERR	:
00000000G	00	05	FB	00049	CALLS	#5, LIB\$SIGNAL	:
		65	DD	00050	PUSHL	LNK\$GL_CURFIL	1214
00000000G	00	01	FB	00052	CALLS	#1, LNK\$CLOSEFILE	:
	63	52	B0	00059	MOVW	SAVEUSZ, LNK\$AL_RAB+32	1215
		09	11	0005C	BRB	2\$	1216
		64	D7	0005E	DECL	HDRBLKCNT	1219
	63	52	B0	00060	MOVW	SAVEUSZ, LNK\$AL_RAB+32	1220
	50	01	D0	00063	MOVL	#1, R0	1221
			04	00066	RET		:
		50	D4	00067	CLRL	R0	1222
			04	00069	RET		:

; Routine Size: 106 bytes, Routine Base: \$CODE\$ + 071D

: 728 1223 1 end
: 729 1224 1
: 730 1225 0 eludom

! End of module

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	16	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	12	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$SPLITS	20	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	1927	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
. ABS .	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	68	0	1000	00:02.0
\$255\$DUA28:[LINKER.OBJ]DATBAS.L32;1	538	41	7	28	00:00.8

LNK_PROCSHRIM
V04=000

K 9
16-Sep-1984 00:30:18
14-Sep-1984 12:40:34

VAX-11 Bliss-32 V4.0-742
[LINKER.SRC]LNKPROSHR.B32;2

Page 24
(3)

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:LNKPROSHR/OBJ=OBJ\$:LNKPROSHR MSRC\$:LNKPROSHR/UPDATE=(ENH\$:LNKPROSHR)

: Size: 1927 code + 48 data bytes
: Run Time: 00:50.4
: Elapsed Time: 02:02.6
: Lines/CPU Min: 1457
: Lexemes/CPU-Min: 28711
: Memory Used: 563 pages
: Compilation Complete

0219 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LNKPROLTB
LIS

LNKSYMTBL
LIS

LNKSYMOUT
LIS

LNKUMALLO
LIS

LNKPSCTBL
LIS

LNKPROSHR
LIS

LNKSTATSD
LIS